Microsoft

.net developer training

Create XML Web Services and Applications with Visual Studio .NET

What is Microsoft® .NET?

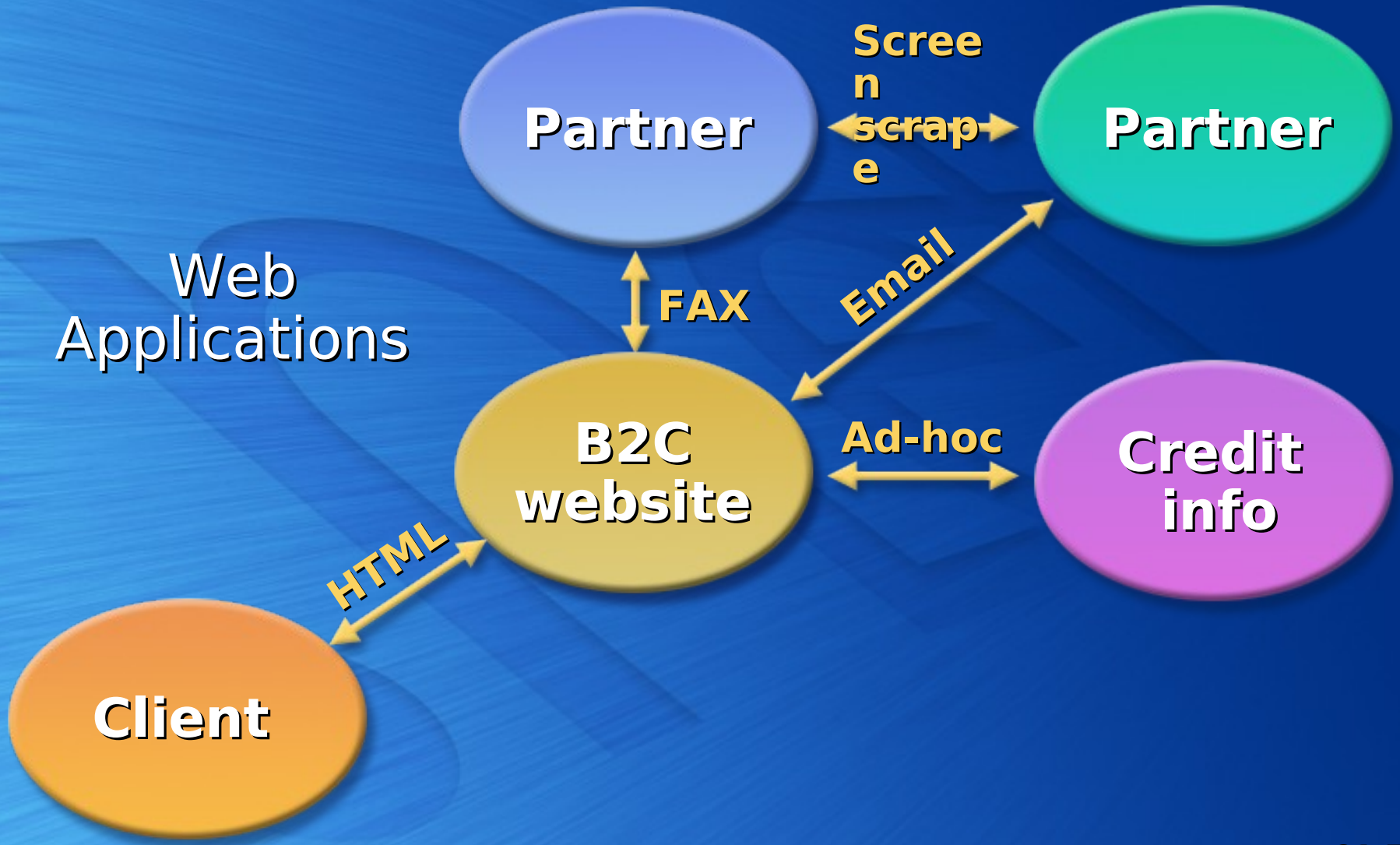
Module 01 Agenda

- Introduction
- XML
- Microsoft .NET
- Inside the .NET Framework

Introduction
Technology Drivers

- Computing power
  - Doubles every 18 months
  - PC-based data centers
- Connectivity
  - Low-cost, broad-reach Internet
  - Wireless, broadband access
- Device proliferation
  - Smart Devices – Personal digital assistants, cell phones and gas pumps

Introduction
Integration Imperative



XML
Lingua Franca for Next Generation Software

A New Generation of Open Software

- Universal data format
- Standard-based integration
- Two-way and adaptive
  - Adapts to devices
  - Adapts sender to receiver
  - User customizable
- Programmable XML Web services

XML

What is An XML Web Service?

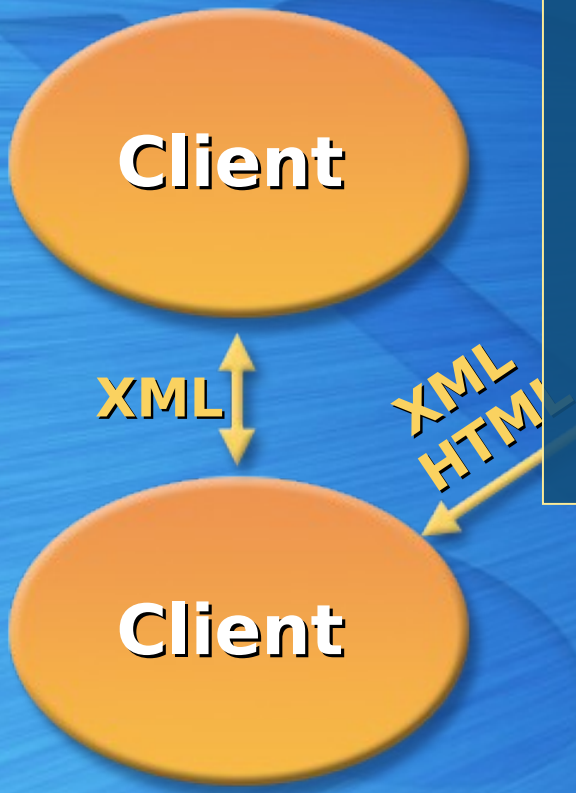
Simple, open, broad support

Publish, find, use services: **UDDI**

Service interactions: **SOAP**

Universal data format: **XML**

Ubiquitous communications: **Internet**



Benefits: **End-to-end automation; Share, analyze and act on the data**

Microsoft .NET
Building on an XML Foundation

Clients

Tools

Web Services

Servers

Experiences & Solutions

Microsoft .NET
Clients

	Today	Future
PCs	Windows 2000 Windows XP	Tablet PC
Smart Devices	Pocket PC, Ultimate TV, Auto PC, Smartcard for Windows , Xbox	N+1 releases “Stinger”
Enablers	Embedded Windows Windows CE Windows CE .NET	Embedded Windows XP

Microsoft .NET
Enterprise Servers

	Today	Future
Windows Servers	Windows 2000 Server Advanced Server Datacenter Server	Windows “Whistler” Server Advanced Server Datacenter Server
.NET Enterprise Servers	SQL Server BizTalk Server Host Integration Server Commerce Server Internet Security and Acceleration Server Application Center Ex…	N+1 Releases Mobile Information Server Supplier Accelerator

Microsoft .net Enterprise Servers

Microsoft Windows 2000 Server

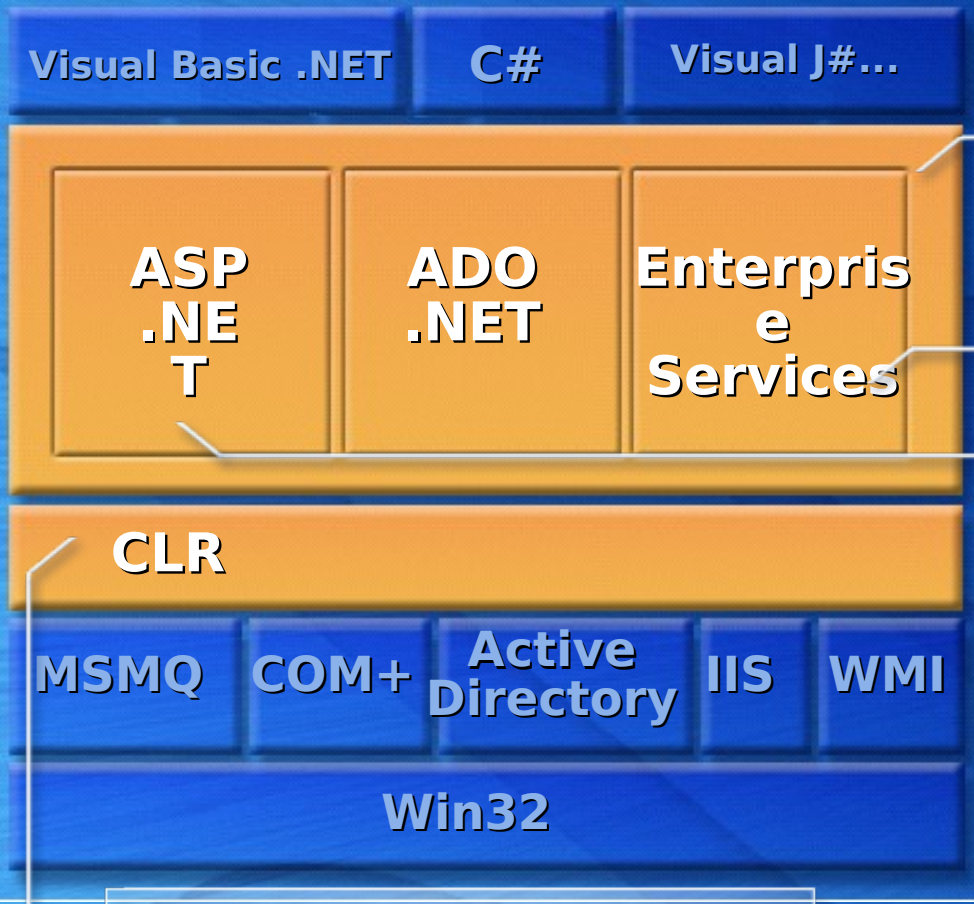
Microsoft .NET
Building Block Services

- **Microsoft Passport**
  - **Authentication service can provide single sign-on capability for any Web site**
  - **More than 160 million-plus active accounts**
- **My Services**

Inside the .NET Framework
Infrastructure for the .NET Platform

- Common Language Runtime
  - Manages running code
  - Verifies type safety
  - Provides garbage collection, error handling
  - Security
  - Provides common type system
  - Provides access to system resources
    - Native API, COM interop, etc.
- A collection of unified classes
  - ASP.NET
  - ADO.NET
  - Enterprise Services
  - System, System.Drawing, System.Web, System.XML, etc.

Inside the .NET Framework
Win32 and the .NET Framework



Unified Class libraries

Unifies Programming models across languages

Factored for extensibility

Designed for tools

Enterprise Services

The services required for mission critical applications: Transactions, Messaging, Partitions, Object Pooling and Events

XML and Data Access

ADO .NET interfaces any database, loosely coupled data access and native XML data format; leverages the huge library of ODBC OLE DB drivers

CLR

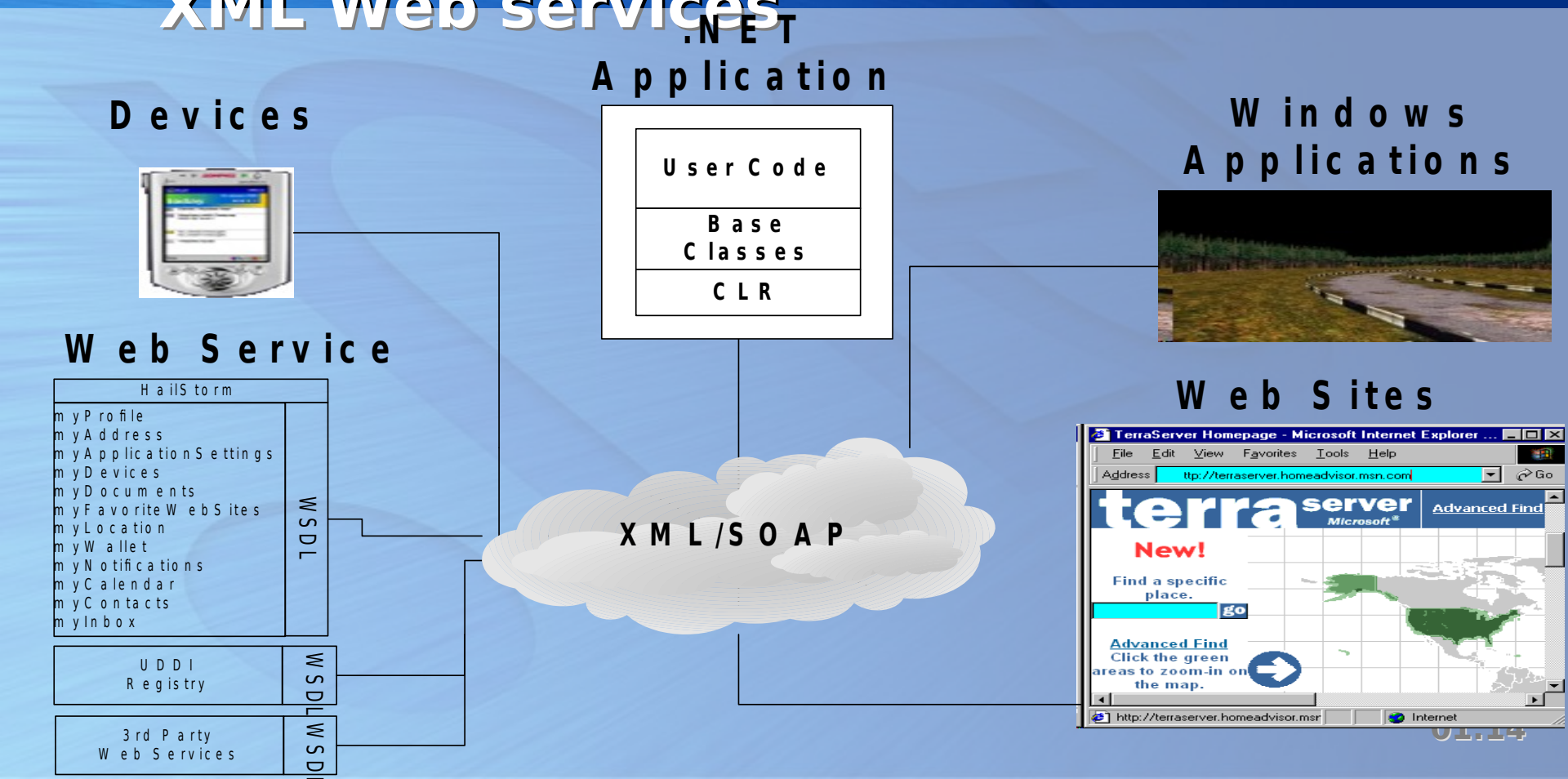
Executes code, maintains security, handles component “plumbing” and dependencies

XML Web Services and Scripting

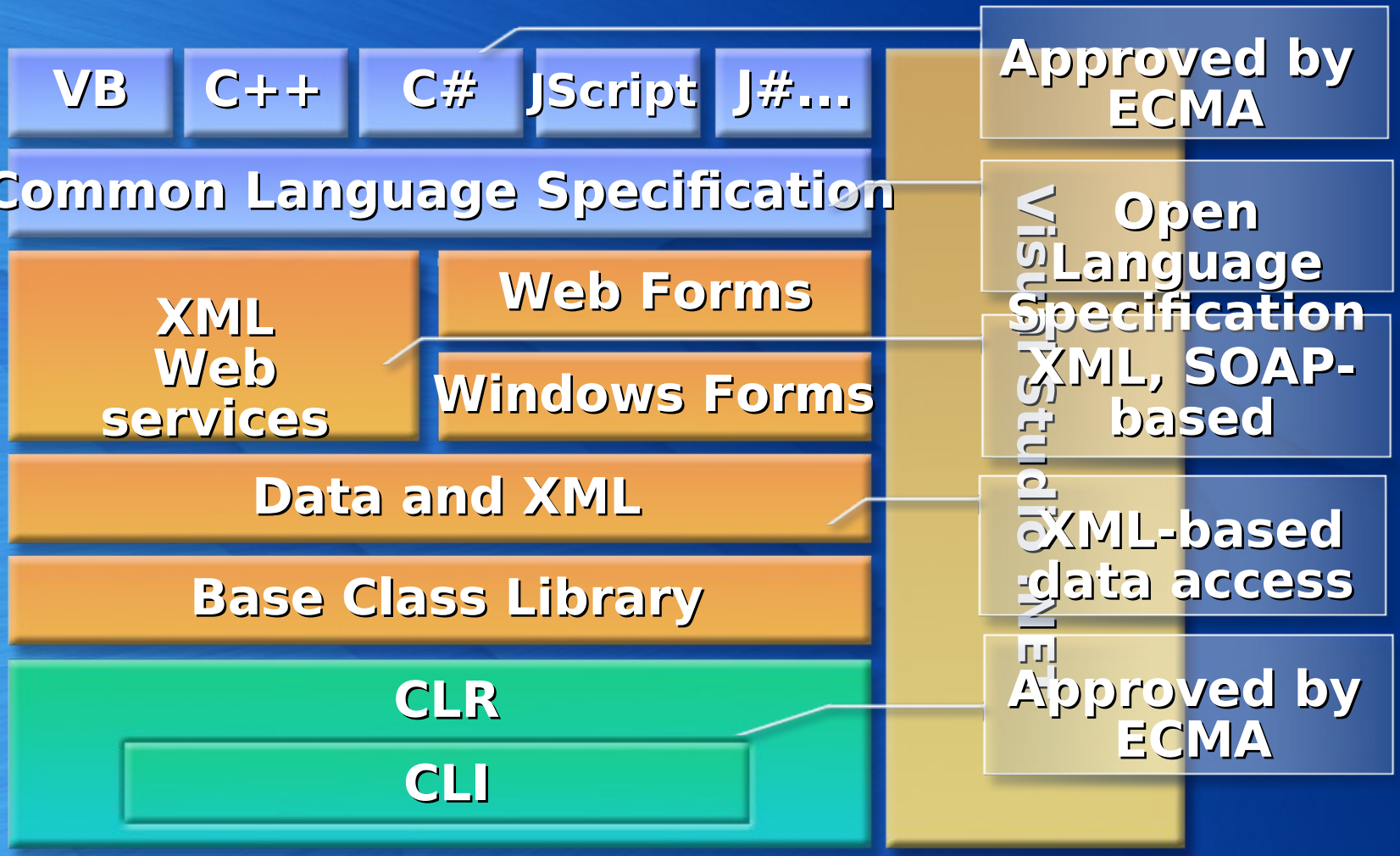
High-productivity environment for building and operating XML Web services

Inside the .NET Framework
Web Services, Windows Applications and more

- .NET Framework favors loosely connected, distributed and stateless XML Web services

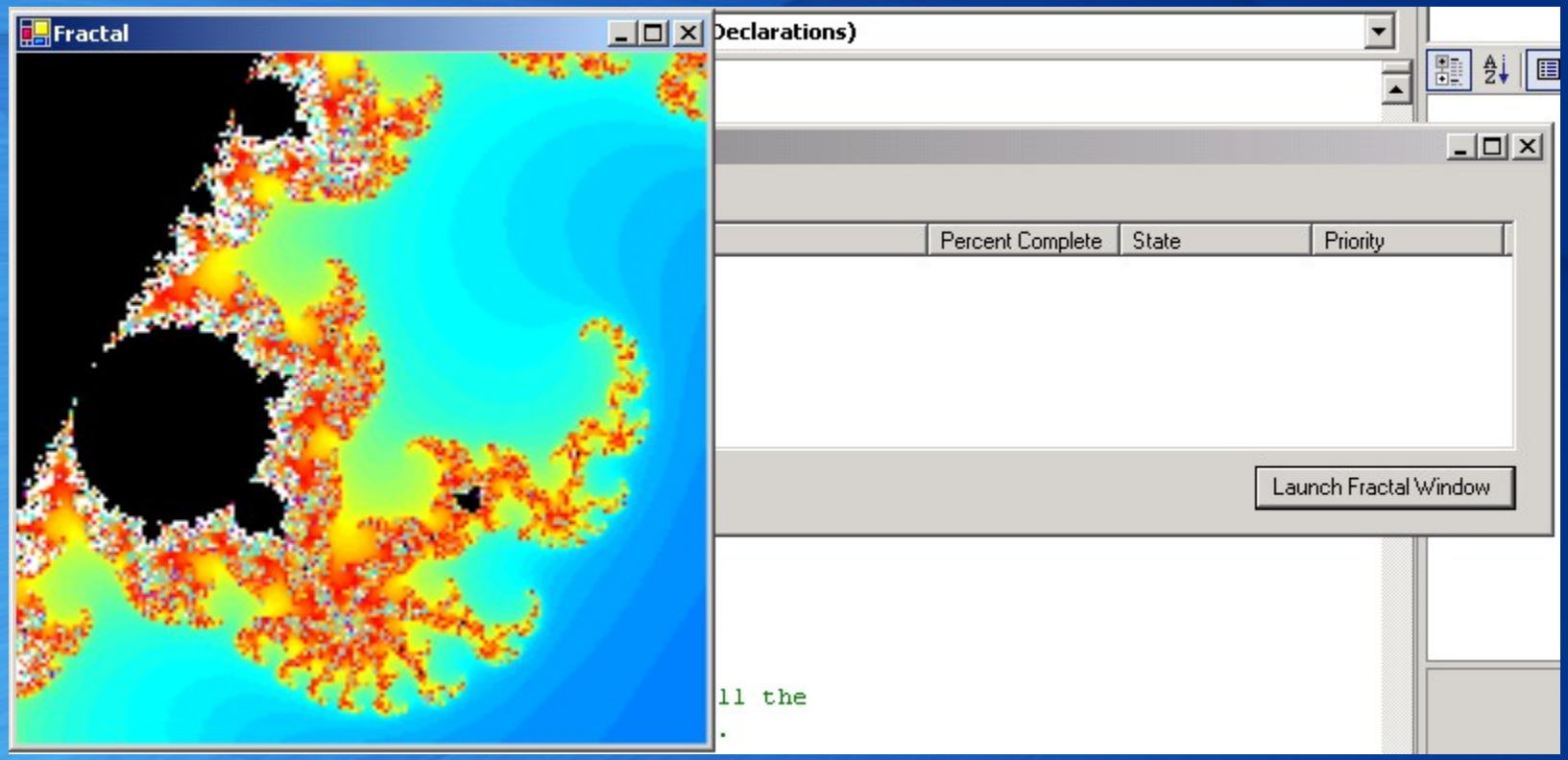


Inside the .NET Framework
Based on Standards



Inside the .NET Framework
Demo

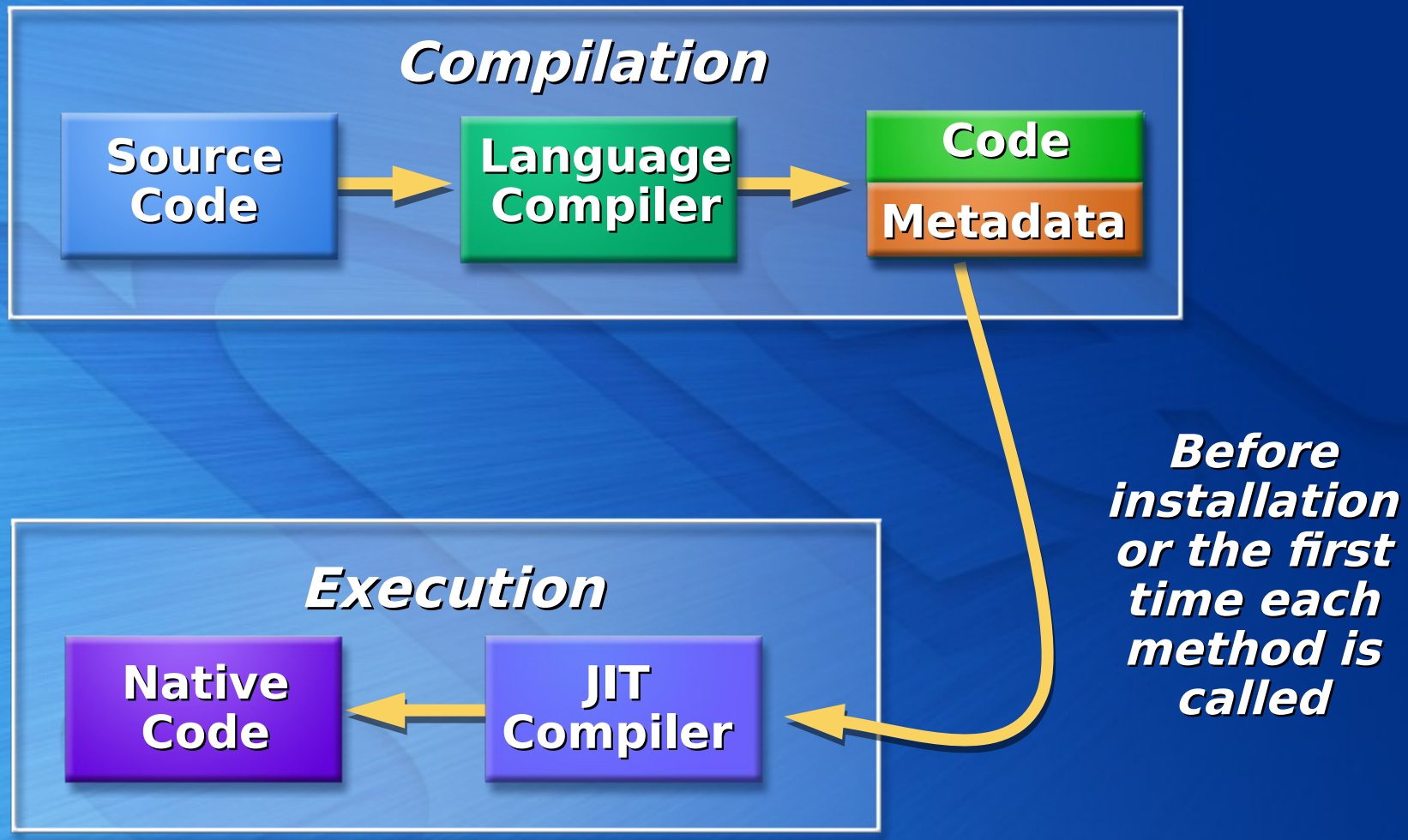
- **Fractals and Threading**



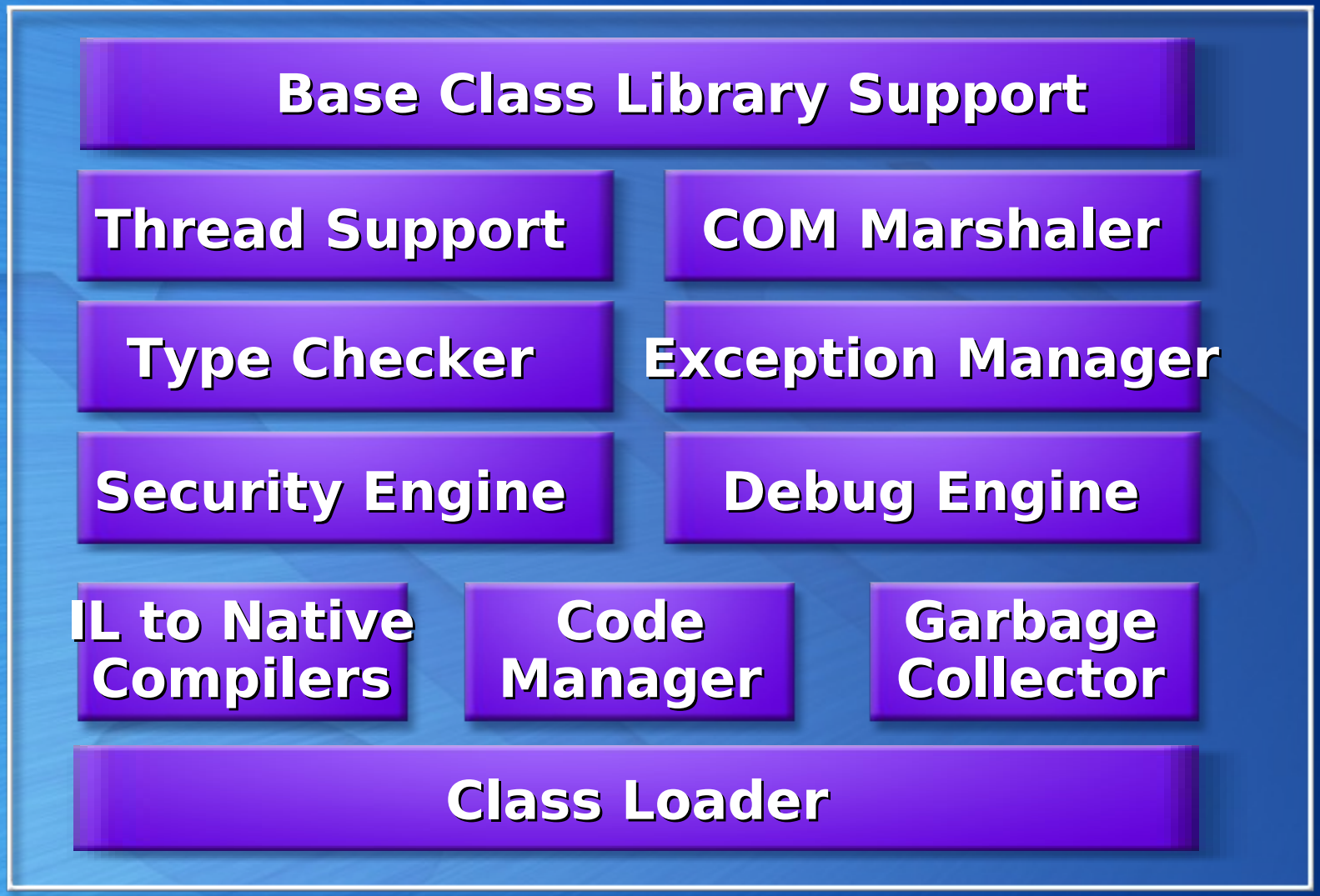
The Common Language Runtime
Features

- Manages running code
  - Threading
  - Memory management
  - No interpreter: Install time or run time IL to native compilation
- Fine-grained evidence-based security
  - Code access security
    - IL can be verified to guarantee type safety
    - No unsafe casts, no un-initialized variables and no out-of-bounds array indexing
  - Role-based security
  - Integrated with underlying OS
- XCOPY deployment
- Object remoting with SOAP

The Common Language Runtime
Compilation and Execution



The Common Language Runtime
Internals



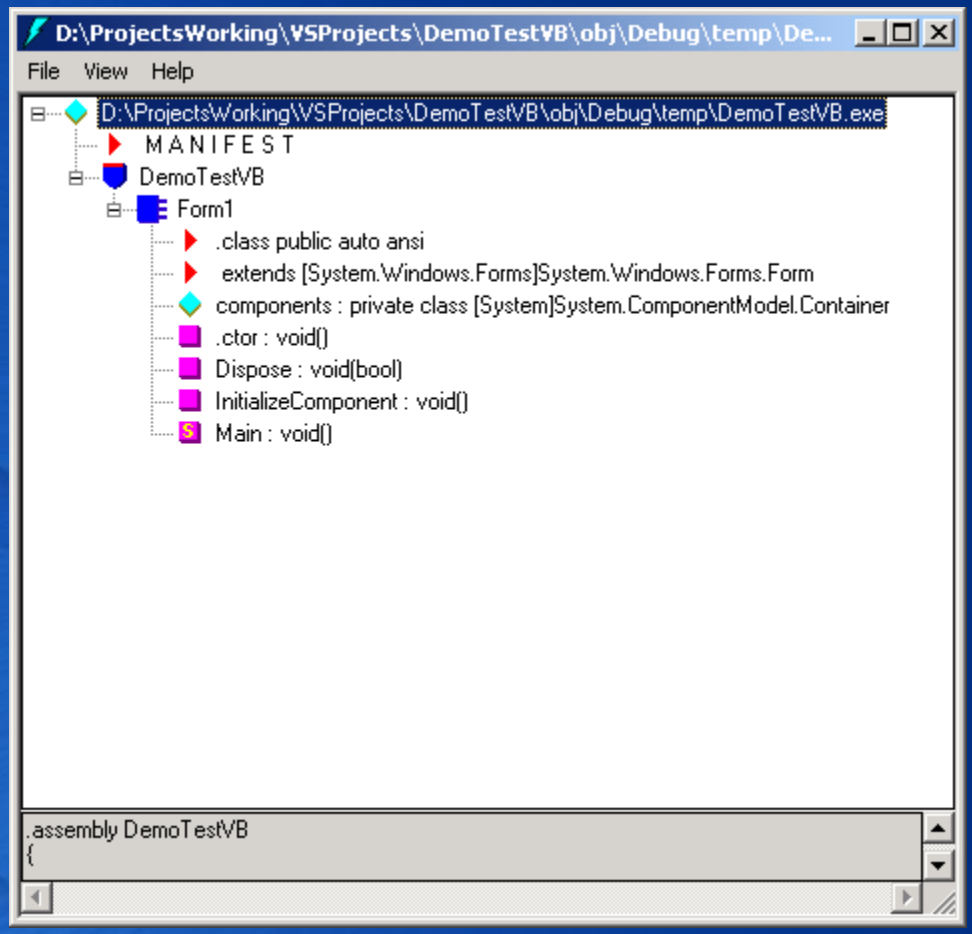
The Common Language Runtime
Simplified Development

- Eliminates COM plumbing
- No more …
  - Registration
  - GUIDs
  - .IDL files
  - HRESULTs
  - IUnknown
  - AddRef/Release
  - CoCreateInstance

=>self describing components
=>hierarchical namespaces
=>source code to metadata
=>structured exceptions
=>root object class
=>garbage collector
=>”new” operator

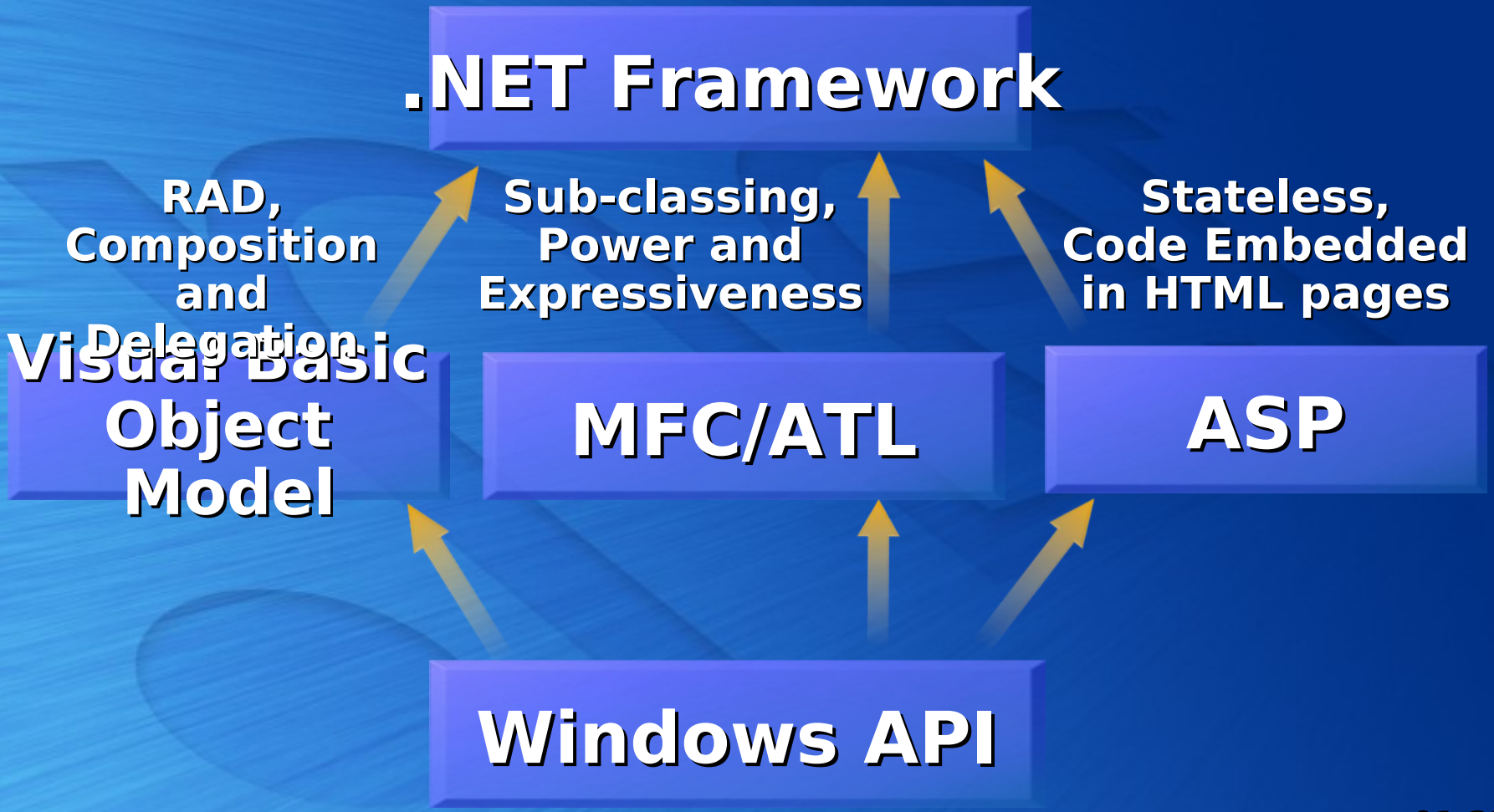
The Common Language Runtime
Demo

- **CLR Basics Using ILDasm**

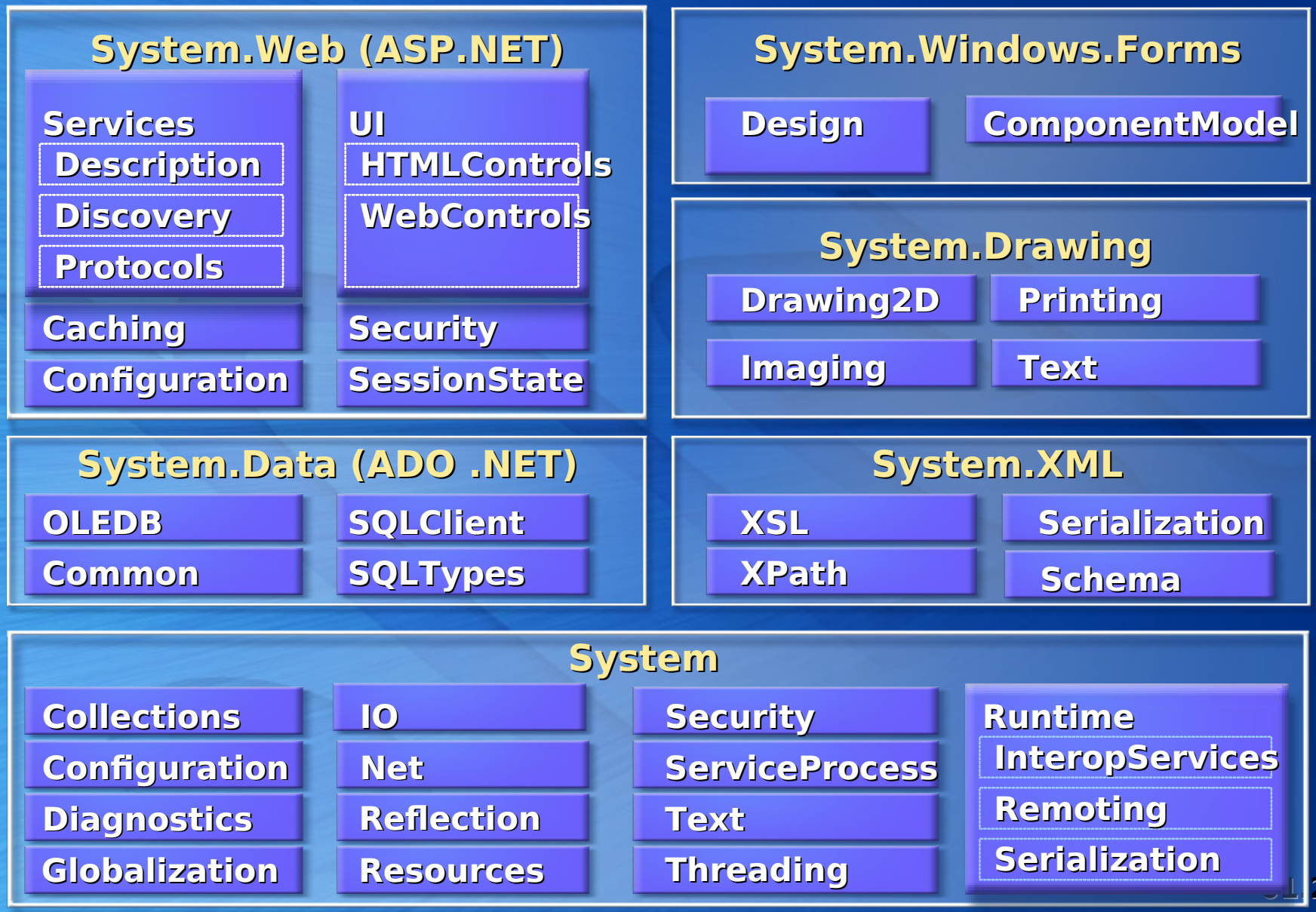


The .NET Unified Programming Model

Consistent API availability regardless of language and programming model



.NET Unified Classes



Multi-Language Development

- Object system is built in, not bolted on
  - Use the language of choice
  - No additional rules or API to learn
- Advanced multi-language features
  - Cross-language inheritance and exceptions
- Supports more than 20 languages
- Tools that work in multiple languages and across languages and platforms
  - Debugger

Simplified Deployment

- **No registration required**
  - Code is self-describing
  - Copy components to application directory
- **Zero impact install**
  - Installing one application fails to affect another
- **Side-by-side execution**
  - Multiple component versions can co-exist

Seamless Integration

- Any .NET class can be a COM object
- COM objects can be imported
  - No ref counting or COM API necessary
- .NET classes use COM+ services
  - Transactions, Object pooling, etc…
- Easily call system functions
  - No marshaling for common cases
- Use XML Web services from Web or Windows applications

Simple to Use

- Component-oriented
  - Properties, methods and events are first class
  - Design-time functionality
- Organization
  - Organized in hierarchical namespaces
- Unified Type System
  - Everything is an object
  - Boxing converts value types to objects
    - No need for special variant type

Summary

- Microsoft .NET
  - Built from the ground up
  - Standards-based platform
  - XML Web services
  - Consistent programming model
- The .NET Platform has four components:
  - .NET Framework and Visual Studio .NET
  - Software for Smart Devices
  - Server infrastructure
  - Building block services
- Common framework for all languages
  - Common Language Runtime and Unified classes

Resources

- *The .NET Home Page* - The best "jump off" point for all things .NET http://www.microsoft.com/net/default.asp
- *Introduction to .NET* - An explanation of the philosophy of and logic behind .NET http://www.microsoft.com/net/intro.asp
- *What is .NET?* Here you will find a basic description of .NET services and functions http://msdn.microsoft.com/theshow/EPISODE011/default.asp
- *Frequently Asked Questions about Microsoft® .NET Framework* - http://msdn.microsoft.com/library/techart/faq111700.htm
- *The GotDotNet Home Page* - An excellent site to find more information about .NET with samples and real-world examples of .NET applications http://www.gotdotnet.com/
- *An Introduction to the .NET Framework* - A concise introduction to the .NET Framework http://www.gotdotnet.com/about_new.aspx

In-Depth Training On The .NET Framework

Direct from Microsoft

- 2349A: Programming the Microsoft .NET Framework with C# (Prerelease)
- 2350A: Securing and Deploying Microsoft .NET Assemblies (Prerelease)
- 2415A: Programming the Microsoft .NET Framework with Visual Basic .NET (Prerelease)

01.32